REMARKS

In the Office Action dated March 31, 2005, claims 1-19 were presented for examination. Claims 1-12, 14-16, 18, and 19 were rejected under 35 U.S.C. §102(e) as being anticipated by *Weinberger et al.*, U.S. Patent No. 6,813,777, and claims 13 and 17 were rejection under 35 U.S.C. §103(a) as being unpatentable over *Weinberger et al.* in view of *Cook*, U.S. Patent No. 4,189,769.

Applicant wishes to thank the Examiner for the careful and thorough review and action on the merits in this application. The following remarks are provided in support of the pending claims and is responsive to the Office Action of March 31, 2005 for the pending application.

I. Interview Summary

On June 28, 2005, Applicant's Attorney conducted a telephonic interview with Examiner Korobov and his Primary Examiner. During the interview, the Examiners indicated their disappointment with the broadness of the claims, and their desire for Applicant to narrow the claims. Applicant's Attorney focused the interview on the element of the Service Processor, and the fact that neither of the prior art references applied by the Examiner include a Service Processor. The Examiners indicated that they are considering any service provided by a unit to a processor as a Service Processor, and gave the example of the audio-visual unit of *Weinberger et al.* as a unit that processes a service to passenger consoles. The Examiners further indicated that they will consider a routing switch equivalent to a Service Processor because it serves data and reports problems to a manager. Applicant's attorney indicated that a Service Processor is a term of art known in the field, and further indicated the audio visual unit of *Weinberger et al.* and a routing switch are not equivalent to a Service Processor.

II. Rejection of claim 1-12, 14-16, 18, and 19 under 35 U.S.C. §102(e)

In the Office Action of March 31, 2005, the Examiner assigned to the application rejected claims 1-12, 14-16, 18, and 19 under 35 U.S.C. §102(e) as being anticipated by *Weinberger et al.*

Weinberger relates to a system that includes a system server and a network supporting multiple computer processors. More specifically, *Weingberger et al.* illustrates a system server where the system's software interfaces with a networked unit server and one or more service clients. This interface is intended to provide services to the client in the form of a passenger entertainment system. There is no teaching in *Weingberger et al.* for utilizing a Service Processor to communicate between a remote console and a quad. A Service Processor is defined as "a computer physically attached to a computer system, wherein the processor's sole function is to control the hardware and provide diagnostic support."[1] In *Weingberger et al.*, the system's software causes a remote console to interface directly with each computer unit to process a particular service to that unit. However, there is no provision in the system of *Weingberger et al.* for utilizing a Service Processor hardware to provide diagnostic support to the server in the event of a requirement for diagnosis or maintenance of server hardware. The Service Processor is "a computer processor device, embedded within another computer system, that can be used to monitor, control, configure, manager, diagnose, or maintain that other larger computer system."[2] In a typical use of a Service Processor, the Service Processor is "logically decoupled from the computer system with which it is associated, such that it can provide information asynchronously, and without relying on the services of the larger system."[3] The fact that *Weinberger et al's* system processes a service for a client does not mean that *Weinberger et al.* discloses the use of Service Processor hardware. "A Service Processor is a separate CPU, and it

[1] See Declaration of Paul E. McKenney Under 37 C.F.R. §1.132,¶ 11, attached as Exhibit A (hereinafter "McKenney Declaration").

[2] See McKenney Declaration, ¶ 9.

[3] See McKenney Declaration, ¶10.

cannot be used for general-purpose computing tasks."[4] In fact, there is no teaching anywhere in the 185 pages of the *Weingberger et al.* patent for a processor whose sole function is to control hardware and provide diagnostic support. Rather, the diagnostic support found in *Weingberger et al.* is limited to existing hardware and software. The Service Processor of Applicant is a separate hardware item that "provides both in-band and out-of-band control and diagnostic support" to a processor that experiences a failure.[5] For example, if a processor in communication with the Service Processor locks up, the Service Processor has the ability to perform diagnostics on the locked up processor to either repair or diagnose the failed processor. The Service Processor of Applicant is not equivalent to a switch or a router with a processor that reports problems to a manager as they occur, nor is the Service Processor of Application equivalent to providing any service to any processor. The Service Processor is an accepted industry term for a processor within a separate computer or computer card that provides control and diagnostic support to any type of service within the system. Accordingly, *Weinberger et al.* does not teach the use of a Service Processor to communicate between a remote console and a quad as claimed by Applicant.

Furthermore, as noted in the preamble of Applicant's claims, 1, 7, 12, and 16, the method and system are operable in a headless environment. "A headless operating system requires that local console input/output dependencies be removed from the operating system, *i.e.* in a headless environment the operating system supports operating without a keyboard, mouse or monitor directly attached to the system." Applicant's Specification, Page 1, lines 18-21. "In computer hardware, 'headless' refers to a server with no monitor attached. Interaction with it depends on the use of a network connection or serial communications. See Exhibit B. "Any terminology in the preamble that limits the structure of the claimed invention must be treated as a claim limitation." MPEP §2111.02, citing *Corning Glass Works v. Sumitomo Elec. U.S.A.*, 868 F.2d 1251, 1257, 9 USPQ 2d 1962 (Fed. Cir. 1989). As noted in claims 1, 7, 12, and 16, each claim preamble states "a computer system operable in a headless environment". This statement was

---

[4] See McKenney Declaration, ¶15.

[5] See McKenney Declaration, ¶12.

placed in the preamble of each of the claims to indicate a statement of the intentional purpose, not as a statement of effect that may or may not be desired. In reviewing the 185 pages of *Weinberger et al.* there is not one reference to the term "headless". Therefore, it is clear that computer system of *Weingberger et al.* is not a system that is operable in a headless environment. As noted in the Background section of Applicant's specification, it is the advent of Microsoft Windows 2000 NT® and a driver associated therewith that prompted the invention of Applicant. Microsoft Windows 2000 NT® was released on February 17, 2000. See Exhibit C. The *Weingberger* et al. patent was filed on May 26, 1998. Accordingly, it is clear that the *Weingberger et al.* patent was not intended to support communication with a computer system operable in a headless environment, as the technology was not available at the time of filing of *Weingberger et al.*

In order for the claimed invention to be anticipated under 35 U.S.C. §102(e), the prior art must teach all claimed limitations presented by the claimed invention. "A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." MPEP §2131 (citing *Verdegaal Bros. v. Union Oil Co. of California*, 814 F. 2d 628, 631, 2 U.S.P.Q. 2d 1051, 1053 (Fed. Cir. 1987)). *Weinberger et al.* does not show all of the elements as claimed by Applicant in pending claims 1-12, 14-16, 18, and 19. Specifically, *Weinberger et al.* does not show the Service Processor, rather *Weinberger* merely shows a system where a remote console interfaces directly with each computer unit and processes service requests by each particular unit. Applicant's system contains a Service Processor for management and hardware and diagnostic support for a remotely located multiprocessing unit. Furthermore, *Weingberger et al.* does not show a computer system operable in a headless environment, as noted in each of Applicant's independent claims. Accordingly, *Weinberger et al.* clearly fails to teach the limitations pertaining to the computer system operable in a headless environment and the Service Processor as presented in Applicant's pending claims 1-12, 14-16, 18, and 19.

Finally, "[a] previous patent anticipates a purported invention only where, except for insubstantial differences, it contains *all* of the same elements operating in the same fashion to

perform an identical function." *Saunders v. Air-Flo Co.*, 646 F.2d 1201, 1203 (7^{th} Cir. 1981) citing *Popeil Brothers, Inc. v. Schick Electric, Inc.*, 494 F. 2d 162, 164 (7^{th} Cir. 1974) (holding patents were not invalid as being anticipated by or obvious in light of prior art) *(emphasis added)*. *Weinberger et al.* does not anticipate the invention of Applicant based upon the legal definition of anticipation. Although the prior art cited by the Examiner relates to a system of a plurality of computers and a remote unit, *Weinberger et al.* fails to show the Service Processor element as presented in Applicant's claimed invention. In fact, *Weinberger et al.* does not show any use or an equivalent use of a Service Processor as defined above. Rather, *Weinberger et al.* shows a system that processes service requests from a user's computer directly to a remote console. Furthermore, *Weinberger et al.* fails to show the headless environment as presented in Applicant's claimed invention. In fact, in the entire 185 pages document of *Weinberger et al.* there is not one use of the term "headless". Accordingly, Applicant respectfully requests the Examiner to remove the rejection of claims 1-19 and to provide allowance of this application.

III. Rejection of claim 13 and 17 under 35 U.S.C. §103(a)

In the Office Action dated March 31, 2005, claims 13 and 17 were rejected under 35 U.S.C. §103(a) as being obvious under *Weinberger et al.*, U.S. Patent No. 6,813,777 in view of *Cook et al.*, U.S. Patent No. 4,189,769.

The remarks pertaining to *Weingberger et al.* provided above are hereby incorporated by reference.

In the Office Action of March 31, 2005, the Examiner stated that the Applicant's use of a UART was obvious under *Weingberger et al.* in view of *Cook et al.* As stated above, *Weingberger et al.* relates to a system that includes a system server and a network supporting multiple computer processors. However, *Weingberger et al.* does not utilize a UART, nor is *Weingberger et al.* operable in a headless environment. *Cook et al.* relates to a system of controlling data transferred between a central processing unit (CPU) and a plurality of peripheral computers. More specifically, the data is transferred from the peripheral computers to the CPU

by a UART multiplexer. However, *Cook et al.*, just like *Weinberger et al.*, is not operable in a headless environment. Accordingly, neither *Cook et al.* nor *Weinberger et al.* teach or discuss utilization of a computer system operable in a headless environment.

"To establish a prima facie case of obviousness . . . the prior art reference (or references when combined) must teach or suggest all the claim limitations. The teaching or suggestion to make the claimed combination and the reasonable expectation of success must both be found in the prior art, and not based on applicant's disclosure." MPEP §2142, citing *In re Vaeck*, 947 F.2d 488, 20 USPQ2d 1438 (Fed. Cir. 1991). As noted in Section I above, *Weinberger et al.* does not teach or discuss a computer system operable in a headless environment. Similarly, *Cook et al.* does not teach this element.

In order to apply *Cook et al.* to Applicant's pending application, *Cook et al.* must be modified and reconfigured to communicate in a computer system operable in a headless environment. However, this is not taught in *Cook et al.* "Although a prior art device 'may be capable of being modified to run the way the apparatus is claimed, there must be a suggestion or motivation in the reference to do so.' " MPEP §2143.01 (citing *In re Mills*, 916 F.2d 680, 682, 16 USPQ 2d. 1430 (Fed. Cir. 1990)). *Cook et al.* does not teach or suggest an ability to communicate in a headless environment. To read *Cook et al.* as providing or supporting the use of a headless communication would require a modification to the invention of *Cook et al.* not envisioned or required in the system of *Cook et al.* The only suggestion for use of a headless environment associated with UART communication channels is derived from Applicant's invention. Absent Applicant's invention, there is no suggestion or motivation within *Cook et al.* for such a modification.

"It is impermissible to use the claimed invention as an instructions manual or 'template' to piece togther the teachings of the prior art so that the claimed invention is rendered obvious." *In re Fritch*, 972 F.2d 1260, 1266, 23 USPQ 2d 1780 (Fed. Cir. 1992) (citing *In re Gorman*, 933 F.2d 982, 987 (Fed. Cir. 1991)). Although Applicant's invention may appear to combine elements found in *Weinberger et al.* and *Cook et al.*, "the inquiry under [35 U.S.C.] §103 is

whether prior use makes the picture of the jigsaw puzzle, rather than its pieces obvious." *Kori Corp. v. Wilco Marsh Buggies & Draglines*, 708 F.2d 151, 155 (5th Cir. 1983). Even with the teachings of *Cook et al.* combined with the teachings of *Weinberger et al.*, there still remains no teaching, suggestion, or motivation for use of communicating through a UART communication channel in a headless environment. The entirety of Applicant's invention is greater than the sum of the parts that comprise the novelty of the invention. "[T]he linchpin is not whether the individual components of the patent were obvious at the time of the invention, but whether the aggregation produced a new or different result or achieved a synergistic effect." *Id.* (citing *Continental Oil co. v. Cole*, 634 F.2d 188, 197 (5th Cir. 1981)). The element that both *Cook et al.* and *Weinberger et al.* fail to address are critical to the advantages found in Applicant's invention. Both *Cook et al.* and *Weinberger et al.* fail to teach all of the claim limitations and fail to establish the prima facie obviousness of the claimed invention anywhere in the specification and associated drawing figures. Accordingly, the Applicant respectfully submits that claims 13 and 17 would not have been obvious in view of *Weinberger et al.* in view of *Cook et al.* and allowance of claims 13 and 17 is respectfully requested.

For the reasons outlined above, withdrawal of the rejection of record and an allowance of claims 1-19 of this application are respectfully requested.

Respectfully submitted,

By: ______________________

Rochelle Lieberman
Registration No. 39,276
Attorney for Applicant

Lieberman & Brandsdorfer, LLC
12221 McDonald Chapel Drive
Gaithersburg, MD 20878-2252
Phone: (301) 948-7775
Fax: (301) 948-7774
Email: rocky@legalplanner.com

Date: June 30, 2005

EXHIBIT

A

PATENT
Attorney Docket No.: BEA9-2001-0036-US1

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of:	Ramey		
SERIAL NO.:	10/040,174	Group Art Unit:	2155
FILING DATE:	January 2, 2002	Examiner:	Korobov, V.
FOR:	**Headless Serial Redirection Through A Service Processor**		

<u>Declaration Of Paul E. McKenney Under 37 C.F.R. §1.132</u>

I, Paul E. McKenney, being hereby warned that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of this document, declare that:

1. I am an employee of International Business Machines Corporation ("IBM"), the assignee of the above captioned patent application.

2. I have a Masters of Science in Computer Science from Oregon State University (with 14 credit-hours of course work from Stanford University) and a doctorate in the field of Computer Science from Oregon Health and Sciences University. I also have a Bachelors of Science in Computer Science and a Bachelors of Science in Mechanical Engineering from Oregon State University.

3. I have been employed by IBM since April 2000 and am now employed by IBM as a Distinguished Engineer. My primary responsibilities include system architecture for the LINUX technology center where I perform various technical and mentoring activities in the areas of symmetric multiprocessing and non-uniform memory access ("NUMA") computing systems.

4. I was employed with Sequent Computer Systems, Inc. from 1990-2000 and served as Chief Technologist for World-Wide Engineering, technical liaison to Intel and a member of the Sequent Patent Committee, and I was employed with SRI International from 1986-1990 and served as the system administrator for their Unix-based timesharing machine and an engineer for packet radio internet gateways.

5. In 2002, I was elected to the IBM Academy of Technology. I am also a member of the System and Technologies Group Software Architecture Board and Advanced eBusiness Council at IBM, and have been named an IBM Distinguished Engineer.

6. I am a named inventor on nineteen U.S. Patent Grants, and twenty-five pending U.S. patent applications. I have published at least eight journal articles, twenty refereed articles for conferences and workshops; eight un-refereed journal articles; three white papers and technical reports; three guest lectures; and one chapter in a book, all in the field of computer science.

7. I have served as a Member End-to-End Task Force within the Internet Activities Board, 1988-1990; a Member of ANSI X3J16 (C++ standards committee) in 1990; a Member of ACM, IEEE, IETF, SAE (Society of Automotive Engineers); and a Member of May 1983 Computer Science delegation to People's Republic of China (Sponsored by the People-to-People Citizen Ambassador Program and delegation led by Dr. Gio Weiderhold of Stanford University). I have also served as Chair of Oregon Chapter of IEEE Computer Society, 1992-1995, Treasurer of Oregon Section of IEEE Computer Society, 1994, and Secretary of Oregon Section of IEEE Computer Society, 1995.

8. I am thoroughly familiar with the structure and function of a Service Processor.

9. To the best of my knowledge, it is generally recognized in the computer science field that the terminology of "service processor" is a computing device, embedded within another computer system, that can be used to monitor, control, configure, manage, diagnose, or maintain that other larger computer system.

10. To the best of my knowledge, it is generally recognized in the computer science field that the service processor is typically logically decoupled from the computer system with which it is associated, such that it can provide information asynchronously, and without relying on the services of the larger system.

11. To the best of my knowledge, it is generally recognized in the computer science field that the service processor is a computer physically attached to a computer system, wherein the processor's sole function is to control the hardware and provide diagnostic support.

12. To the best of my knowledge, it is generally recognized in the computer science field that the service processor provides in-band and out-of-band control and diagnostic support.

13. To the best of my knowledge, a configuration and functionality of a router is not equivalent to that of a service processor, and the router does not have the capability of meeting the support and diagnostic services of the service processor.

14. To the best of my knowledge, a configuration and functionality of a switch is not equivalent to that of a service processor, and the switch does not have the capability of meeting the support and diagnostic services of the service processor.

15. To the best of my knowledge, a Service Processor is a separate CPU, and it cannot be used for general-purpose computing tasks.

I, Paul E. McKenney, declare under penalty of perjury that all statements made of my own knowledge are true and all statements made on information and belief are believed to be true.

DATE: 30-JUNE-2005

Paul E. McKenney

EXHIBIT
B

Answers.com

Business Entertainment Games Health People Places Reference Science Shopping New Words More...

headless

On this page:
Dictionary

Dictionary

HOUGHTON MIFFLIN COMPANY

head·less (hĕd'lĭs)

adj.

1. a. Formed without a head.
   b. Decapitated.
2. Lacking a leader or director.
3. Lacking intelligence and prudence; stupid or foolish.

head'less·ness *n.*

WordNet

Note: click on a word meaning below to see its connections and related words.

The *adjective* headless has 2 meanings:

Meaning #1: not having a head or formed without a head
 Antonym: headed (meaning #3)

Meaning #2: not using intelligence
 Synonym: brainless

Wikipedia

headless

In software, 'headless' refers to computer programs that use textual input/output to interact with users, instead of using graphics or graphical user interfaces (GUIs).

In computer hardware, 'headless' refers to a server with no monitor attached. Interaction with it depends on the use of a network connection or serial communications.

A 'headless' is also a type of foe encountered in the role-playing universe of Ultima.

This entry is from Wikipedia, the leading user-contributed encyclopedia. It may not have been reviewed by professional editors (see full disclaimer)

Mentioned In

headless is mentioned in the following topics:

acephalous
Headless Cross
Melling, Merseyside
Freedom Park
Sleepy Hollow (movie)
"The Legend of Sleepy Hollow" (Literature)
Dullahan
Theater of Salvation
Glen Phillips
Chris Holmes
More>

Copyrights:

EXHIBIT
C

Answers.com Go

Windows 2000

On this page:
Technology

Technology
Windows 2000

Computer Desktop Encyclopedia

Also called "Win2K" and "W2K," Windows 2000 was a major upgrade to Windows NT 4, launched in early 2000. Available in one client and three server versions, Windows 2000 added support for Plug and Play. Windows 2000 uses the same interface as Windows 95/98, but added considerably more features, dialogs and options.

From NT Domains to Active Directory
Windows 2000 supports Active Directory, which replaces NT's domain system and makes network administration simpler. This is a major redesign of the directory structure for companies. Windows 2000 is more stable than NT and is designed to eliminate erroneous replacement of DLLs when applications are installed (see DLL hell).

Versions

Windows 2000 Advanced Server is similar to Windows NT Server, Enterprise Edition, which supports clustering and automatic failover in the event of a system failure. Windows 2000 DataCenter Server supports more advanced clustering and is the top server offering. Windows 2000 Professional is the client version. See Windows, Windows Server 2003, Windows NT, Windows XP, Active Directory and Plug and Play.

Windows 2000 Version	Use	SMP Support	RAM
Professional	Client		2GB
Server	Server	4-way	2GB
Advanced Server*	Server	8-way	8GB
DataCenter Server*	Server	32-way	64GB

*Supports clustering, failover and load balancing

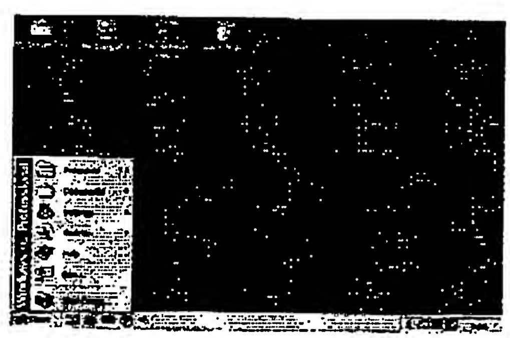
Wikipedia
Windows 2000

Microsoft Windows 2000 (also referred to as **Win2K** or **Windows NT 5.0**) is a 32-bit graphical business-oriented operating system released on February 17, 2000 by Microsoft. Windows 2000 comes in four versions: Professional, Server, Advanced Server, and Datacenter Server. Additionally, Microsoft offers Windows 2000 Advanced Server, Limited Edition, released in 2001, which runs on Intel Itanium 64-bit processors.

Microsoft has replaced Windows 2000 Server products with Windows Server 2003, and Windows 2000 Professional with Windows XP Professional.

Windows Neptune started development in 1999, and was supposed to be the home-user edition of Windows 2000. However, the project lagged in production time - and only one alpha release was built.

Windows 2000



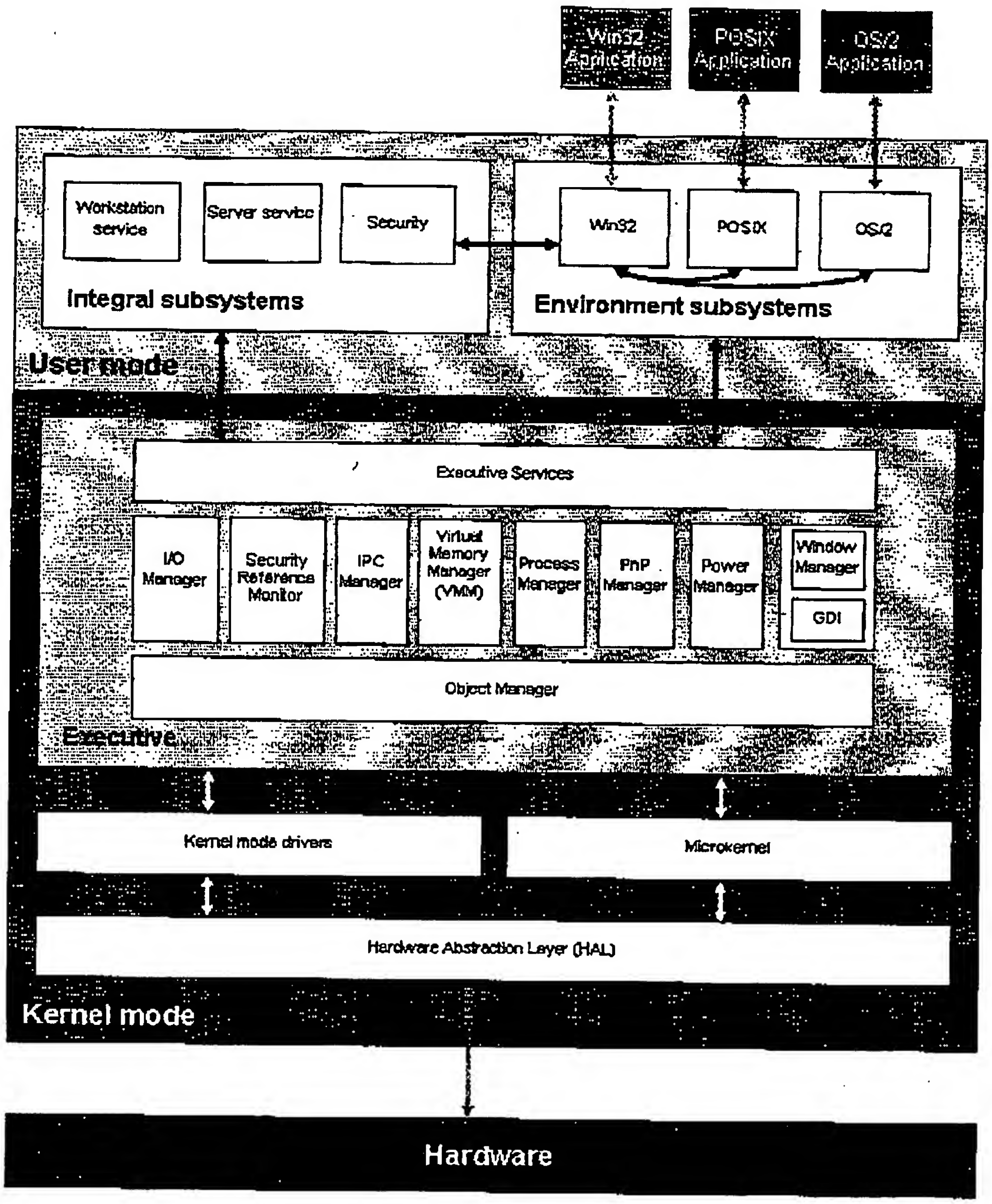
Windows Me was released as a substitute, and the Neptune project was forwarded to the production of Whistler (Windows XP).

Windows 2000 succeeded Windows NT 4.

Developer	Microsoft
OS family	Windows NT
Source model	Closed source
Latest release	Service Pack 4 / FIXME
Kernel type	Hybrid kernel
License	Microsoft EULA
Working state	Historic, but still supported
Website	www.microsoft.com/windows2000

Architecture

Windows 2000 is a 32-bit, preemptible, interruptible operating system, which has been designed to work with either uniprocessor or symmetrical multi processor (SMP) based Intel x86 computers. To process I/O requests it uses packet driven I/O which utilise I/O request packets (IRPs) and asynchronous I/O. It is a highly modular system and, as with most other monolithic operating systems, consists of two main layers: a user mode and a kernel mode. However, Windows 2000 is known as a hybrid operating system as the microkernel is essentially the kernel, while higher-level services are implemented by the executive.



The Windows 2000 operating system architecture consists of two layers (user mode and kernel mode), with many different modules within both of these layers.

User mode

The user mode is made up of subsystems which can pass I/O requests to the appropriate kernel mode drivers via the I/O manager (which exists in kernel mode). Two subsystems make up the user mode layer of Windows 2000: the *Environment subsystem* and the *Integral subsystem*.

Environment subsystem

The Environment subsystem was designed to run applications written for many different types of operating systems. None of the environment subsystems can directly access hardware, and must request access to memory resources through the Virtual Memory Manager that runs in kernel mode. Also, applications run at a lower priority than kernel mode processes. Currently, there are three main environment subsystems: the Win32 subsystem, an OS/2 subsystem and a POSIX subsystem.

The Win32 subsystem can run 32-bit Windows applications. It contains the console as well as text window support, shutdown and hard-error handling for all other environment subsystems. It also supports Virtual DOS Machines (VDMs), which allow MS-DOS and 16-bit Windows 3.x (Win16) applications to be run on Windows. There is a specific MS-DOS VDM which runs in its own address space and which emulates an Intel 486 running MS-DOS 5. Win16 programs, however, run in a Win16 VDM. Each program, by default, runs in the same process, thus using the same address space, and the Win16 VDM gives each process it's own thread to run on. However, Windows 2000 does allow users to run a Win16 program in a seperate Win16 VDM, which allows the program to be preemptively multitasked as Windows 2000 will preempt the whole VDM process, which only contains one running application.

The OS/2 subsystem supports 16-bit character-based OS/2 applications and emulates OS/2 1.3 and 1.x, but not 2.x or later OS/2 applications. The POSIX subsystem supports applications that are strictly written to either the POSIX.1 standard or the related ISO/IEC standards.

Integral subsystem

The Integral subsystem looks after operating system specific functions on behalf of the environment subsystem. It consists of a *security subsystem*, a *workstation service* and a *server service*. The security subsystem deals with security tokens, grants or denies access to user accounts based on resource permissions, handles logon requests and initiates logon authentication, and determines which system resources need to be audited by Windows 2000. It also looks after Active Directory. The workstation service is an API to the network redirector, which provides the computer access to the network. The server service is an API that allows the computer to provide network services.

Kernel mode

Windows 2000 kernel mode has full access to the hardware and system resources of the computer and runs code in a protected memory area. It controls access to scheduling, thread prioritisation, memory management and the interaction with hardware. The kernel mode stops user mode services and applications from accessing critical areas of the operating system that they should not have access to as user mode processes ask the kernel mode to perform such operations on it's behalf.

Kernel mode consists of *executive services*, which is itself made up on many modules that do specific tasks, *kernel drivers*, a *microkernel* and a *Hardware Abstraction Layer*, or HAL.

Executive

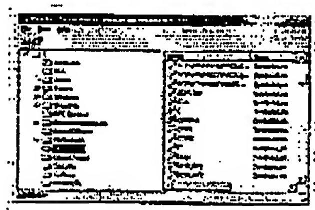
The Executive interfaces with all the user mode subsystems. It deals with I/O, object management, security and process management. It contains various components, including the *I/O Manager*, the *Security Reference Monitor*, the *IPC Manager*, the *Virtual Memory Manager* (VMM), a *PnP Manager* and *Power Manager*, as well as a *Window Manager* which works in conjunction with the Windows *Graphical Device Interface* (GDI). Each of these components exports a kernel-only support routine allows other components to communicate with one another. Grouped together, the components can be called *executive services*. No executive component has access to the internal routines of any other executive component.

- I/O Manager: allows devices to communicate with user-mode subsystems. It translates user-mode read and write commands in read or write IRPs which it passes to device drivers. It accepts file system I/O requests and translates them into device specific calls, and can incorporate low-level device drivers that directly manipulate hardware to either read input or write output. It also includes a cache manager to improve disk performance by caching read requests and write to the disk in the background
- Security Reference Monitor (SRM): the is the primary authority for enforcing the security rules of the security integral subsystem [1]. It determines whether an object or resource can be accessed, via the use of access control lists (ACLs), which are themselves made up of access control entries (ACEs). ACEs contain a security identifier (SID) and a list of operations that the ACE gives a select group of trustees — a user account, group account, or logon session [2] — permission (allow, deny, or audit) to that resource. [3] [4]
- IPC Manager: the IPC manager (or Interprocess Communication Manager) manages the communication between clients (the environment subsystem) and servers (components of the Executive). It has two facilities that it can use: the *Local Procedure Call* (LPC) facility (clients and servers on the one computer) and the *Remote Procedure Call* (RPC) facility (where clients and servers are situated on different computers. Microsoft has had significant security issues with the

RPC facility [5].

- Virtual Memory Manager: manages virtual memory, allowing Windows 2000 to use the hard disk as a primary storage device (although strictly speaking it is secondary storage). It controls the paging of memory in and out of physical memory to disk storage.
- Process Manager: handles process and thread creation and termination
- PnP Manager: handles Plug and Play and supports device detection and installation at boot time. It also has the responsibility to stop and start devices on demand – sometimes this happens when a bus gains a new device and needs to have a device driver loaded to support that device. Both Firewire and USB are hot-swappable and require the services of the PnP Manager to load, stop and start devices. The PnP manager interfaces with the HAL, the rest of the executive (as necessary) and with device drivers.
- Power Manager: the power manager deals with power events and generates power IRPs. It coordinates these power events when several devices send a request to be turned off it determines the best way of doing this.
- The display system has been moved from user mode into the kernel mode as a device driver contained in the file *Win32k.sys*. There are two components in this device driver – the Window Manager and the GDI:
  - Window Manager: responsible for drawing windows and menus. It controls the way that output is painted to the screen and handles input events (such as from the keyboard and mouse), then passes messages to the applications that need to receive this input
  - GDI: the graphical device interface is responsible for tasks such as drawing lines and curves, rendering fonts and handling palettes. Windows 2000 introduced native alpha blending into the GDI.

Object manager



Each object in Windows 2000 exists in it's own namespace. This is a screenshot from *SysInternal's* WinObj.

The *Object manager* is a special executive subsystem that all other executive subsystems must pass through to gain access to Windows 2000 resources – essentially making it a resource management infrastructure service. The object manager is used to reduce the duplication of object resource management functionality in other executive subsystems, which could potentially lead to bugs and make development of Windows 2000 harder [6]. To the object manager, each resource is an object, whether that resource is a physical resource (such as a filesystem or peripheral) or a logical resource (such as a file). Each object has a structure or *object type* that the object manager must know about. When another executive subsystem requests the creation of an object, they send that request to the object manager which creates an empty object structure which the requesting executive subsystem then fills in [7]. Object types define the object procedures and any data specific to the object. In this way, the object manager allows Windows 2000 to be a object oriented operating system, as object types can be thought of as classes that define objects.

Each instance of an object that is created stores it's name, parameters that are passed to the object creation function, security attributes and a pointer to it's object type. The object also contains a object close procedure and a reference count to tell the object manager how many other objects in the system reference that object and thereby determines whether the object can be destroyed when a close request is sent to it [8].

Every object exists in a hierachical object namespace.

Microkernel

The Microkernel sits between the HAL and the Executive and provide multiprocessor synchronization, thread and interrupt scheduling and dispatching, and trap handling and exception dispatching. The Microkernel often interfaces with the process manager. [9] The microkernel is also responsible for initialising device drivers at bootup that are necessary to get the operating system up and running.

Kernel-mode drivers

Windows 2000 uses kernel-mode device drivers to enables the Windows 2000 to interact with hardware devices. Each of the drivers has well defined system routines and internal routines that it exports to the rest of the operating system. All devices are seen by user mode code as a file object in the I/O manager, though to the I/O manager itself the devices are seen as device objects, which it defines as either file, device or driver objects. Kernel mode drivers exist in three levels: highest level drivers, intermediate drivers and low level drivers. The highest level drivers, such as file system drivers for FAT and NTFS, rely on intermediate drivers. Intermediate drivers consist of function drivers – or main driver for a device – that are optionally sandwiched between lower and higher level filter drivers. The function driver then relies on a bus driver – or a driver that services a bus controller, adapter, or bridge – which can have an optional bus filter driver that sits between itself

and the function driver. Intermeditate drivers rely on the lowest level drivers to function. The Windows Driver Model (WDM) exists in the intermediate layer. The lowest level drivers are either legacy Windows NT device drivers that control a device directly or can be a PnP hardware bus. These lower level drivers directly control hardware and do not rely on any other drivers.

Windows Driver Model

Windows 2000 introduced the Windows Driver Model (WDM) driver model to the NT kernel. WDM exists in the intermediary layer of Windows 2000 kernel-mode drivers and was introduced to increase the functionality and ease of writing drivers for Windows. The WDM was mainly designed to be binary and source compatible between Windows 98 and Windows 2000. However, this may not always be desired and so specific drivers can be developed for either operating system. WDM consists of:

- Class drivers: these can be thought of as built-in *framework* drivers that miniport and other class drivers can be built on top of. The class drivers provide an interfaces between different levels of the WDM architecture. Common functionality between different classes of drivers can be written into the class driver and used by other class and miniport drivers. The lower edge of the class driver will have its interface exposed to the miniport driver, while the upper edge of top level class drivers is operating system specific. Class drivers can be dynamically loaded and unloaded at will. They can do class specific functions that are not hardware or bus-specific (with the exception of bus-type class drivers) and in fact sometimes only do class specific functions like enumeration).
- **Miniport drivers:** these are USB, Audio, SCSI and network adapters. They should usually be source and binary compatible between Windows 98 and Windows 2000 and are hardware specific but control access to the hardware through a specific bus class driver.
- **Software bus drivers:** Microsoft provides bus drivers for most common buses, such as PCI, PnpISA, SCSI, USB and Firewire. Each software vendor can create their own bus drivers if needed.
- OS Services: this layer is all the operating system functionality that has been abstracted away from the miniport driver.
- **Virtualisation drivers:** have been part of Windows since v3.0 and are used for legacy hardware.

In the layered architecture of Windows kernel-mode drivers, class/mini port drivers are functional drivers.

Hardware Abstraction Layer

The Windows 2000 Hardware Abstraction Layer, or HAL, is a layer between the physical hardware of the computer and the rest of the operating system. It was designed to hide differences in hardware and therefore provide a consistent platform to run applications on. The HAL includes hardware specific code that controls I/O interfaces, interrupt controllers and multiple processors.

Windows 2000 used to support the DEC Alpha, however they did not extend Alpha support beyond beta 3 of Windows 2000. The HAL now only supports hardware that is compatible with the Intel x86 architecture.

Windows 2000 Core Features

All versions of Windows 2000 share certain features.

NTFS5

Version 3 of the NTFS, (also known as version 5.0), introduced quotas, file-system-level encryption (called EFS), sparse streams and reparse points, which are used to implement Directory Junctions, Volume Mount Points, Hierarchical Storage Management, Native Structured Storage and Single Instance Storage. By adding these features, Windows could compete with established file serving systems like Netware and Unix.

Encrypting File System

The Encrypting File System (EFS) introduced strong encryption into the Windows file world. It allowed any folder or drive to be encrypted and was transparent once implemented. As of February 2004, its encryption has not been compromised.

Versions

Windows 2000 Professional

Windows 2000 Professional is designed as a desktop operating system in business environments. It offers greater security and stability than previous Windows desktop operating systems. It supports up to two processors, and can address up to 4GB of RAM.

Windows 2000 Server

The various server products share the same user interface with Windows 2000 Professional, but contain additional components for running infrastructure and application software. A significant component of the server products is Active Directory, which is an enterprise-wide directory service based on LDAP. Additionally, Microsoft integrated Kerberos network authentication, replacing the often-criticised NT 4 authentication system. This also provided a purely transitive-trust relationship between Windows 2000 domains in a 'forest' (a collection of one or more Windows 2000 domains that share a common schema, configuration, and global catalog, being linked with two-way transitive trusts). Furthermore, Windows 2000 introduced a DNS server which allows dynamic registration of IP addresses.

Windows 2000 Advanced Server

Windows 2000 Advanced Server is a variant of Windows 2000 Server operating system designed for medium-to-large businesses.

A limited edition 64 bit version of Windows 2000 Advanced Server was made available via the OEM Channel.

Windows 2000 Datacenter Server

Windows 2000 Datacenter Server is a variant of the Windows 2000 Server that is designed for large businesses that move large quantites of confidential or sensitive data frequently via a central server.

Its system requirements are normal, but is compatible with vast amounts of power:

- A Pentium-class CPU at 400 MHz or higher - up to 32 are supported in one machine
- 256MB of RAM - up to 64GB is supported in one machine
- Approximitely 1GB of available disk space

Total Cost of Ownership

Microsoft commissioned a firm to determine the total cost of ownership (TCO) for enterprise applications on Windows 2000, such as security and other infrastructure tasks, and Web Serving. Windows 2000 had a lower TCO for the four infrastructure items (according to the report), but Linux had a lower TCO for web serving. There has been a lot of controversy over this claim, including:

- Claims that the test were done on different spec machines to give Microsoft an unfair advantage
- Claims that as Microsoft was paying for the report, the neutrality of it is in question

Criticisms

One aspect of concern with Windows 2000 (along with previous versions of NT), is the lack of an option to make a bootable DOS diskette. Unlike previous versions of Windows, which are based on DOS, (Windows 95, Windows 98, Windows Me), when running Windows 2000, a user is unable to make a bootable DOS diskette. While this is not a major issue for the average user, there are times when a DOS boot diskette is required (such as when doing a BIOS upgrade). In instances such as that, some users have turned to alternative sources for boot diskettes, such as BootDisk.com.

An alternative to the bootable diskette is the Recovery Console. As diskettes are rapidly becoming obsolete, the main alternate boot device is the CD-ROM drive; users can access the Recovery Console when booting the install disc. The Recovery Console provides basic command-line functionality, including additional commands to enable and disable Windows services, among other things. The Recovery Console can also be installed onto an existing Windows 2000 installation to appear as an option on boot-up, making it easier to use than having to boot from a CD-ROM drive, but this isn't well documented by Microsoft.

Windows NT also introduced permissions for Registry editing. Windows 2000 incorporated both the Windows 9x REGEDIT.EXE program and NT's REGEDT32.EXE program. REGEDIT.EXE had a left-side tree view that began at "My Computer" and listed all loaded hives. REGEDT32.EXE had a left-side tree view, but each hive had its own window, so the tree displayed only keys. REGEDIT.EXE represented the three components of a value (its name, type, and data) as separate columns of a table. REGEDT32.EXE represented them as a list of strings. REGEDIT.EXE was written for the Win32 API and supported right-clicking

of entries in a tree view to adjust properties and other settings. REGEDT32.EXE was written for the NT 3.x API and required all actions to be performed from the top menu bar. Because REGEDIT.EXE was directly ported from Windows 98, it did not support permission editing (permissions do not exist on Windows 9x). Therefore, the only way to access the full functionality of an NT registry was with REGEDT32.EXE, which many considered to be inefficient and out-of-date. Windows XP was the first system to integrate these two programs into one, adopting the REGEDIT.EXE behavior with the additional NT functionality.

Notes

1. ^ *Microsoft*. Active Directory Data Storage.
2. ^ *MSDN*. Trustee definition.
3. ^ Siyan, Kanajit S., 2000.
4. ^ *MSDN*. ACE definition.
5. ^ Microsoft has had numerous security issues caused by vulnerabilities in its RPC mechanisms. A list follows of the security bulletins that Microsoft have issued in regards to RPC vulnerabilities:
   - Microsoft Security Bulletin MS03-026: issue with a vulnerability in the part of RPC that deals with message exchange over TCP/IP. The failure results because of incorrect handling of malformed messages. This particular vulnerability affects a Distributed Component Object Model (DCOM) interface with RPC, which listens on RPC enabled ports.
   - Microsoft Security Bulletin MS03-001: A security vulnerability results from an unchecked buffer in the Locator service. By sending a specially malformed request to the Locator service, an attacker could cause the Locator service to fail, or to run code of the attacker's choice on the system.
   - Microsoft Security Bulletin MS03-026: Buffer overrun in RPC may allow code execution
   - Microsoft Security Bulletin MS03-010: This particular vulnerabilty affects the RPC Endpoint Mapper process, which listens on TCP/IP port 135. The RPC endpoint mapper allows RPC clients to determine the port number currently assigned to a particular RPC service. To exploit this vulnerability, an attacker would need to establish a TCP/IP connection to the Endpoint Mapper process on a remote machine. Once the connection was established, the attacker would begin the RPC connection negotiation before transmitting a malformed message. At this point, the process on the remote machine would fail. The RPC Endpoint Mapper process is responsible for maintaining the connection information for all of the processes on that machine using RPC. Because the Endpoint Mapper runs within the RPC service itself, exploiting this vulnerability would cause the RPC service to fail, with the attendant loss of any RPC-based services the server offers, as well as potential loss of some COM functions.
   - Microsoft Security Bulletin MS04-029: This RPC Runtime library vulnerability was addressed in CAN-2004-0569, however the title is "Vulnerability in RPC Runtime Library Could Allow Information Disclosure and Denial of Service".
   - Microsoft Security Bulletin (MS00-066): A remote denial of service vulnerability in RPC is found. Blocking ports 135-139 and 445 can stop attacks.
   - Microsoft Security Bulletin MS03-039: "There are three newly identified vulnerabilities in the part of RPCSS Service that deals with RPC messages for DCOM activation- two that could allow arbitrary code execution and one that could result in a denial of service. The flaws result from incorrect handling of malformed messages. These particular vulnerabilities affect the Distributed Component Object Model (DCOM) interface within the RPCSS Service. This interface handles DCOM object activation requests that are sent from one machine to another. An attacker who successfully exploited these vulnerabilities could be able to run code with Local System privileges on an affected system, or could cause the RPCSS Service to fail. The attacker could then be able to take any action on the system, including installing programs, viewing, changing or deleting data, or creating new accounts with full privileges. To exploit these vulnerabilities, an attacker could create a program to send a malformed RPC message to a vulnerable system targeting the RPCSS Service."
   - Microsoft Security Bulletin MS01-041: "Several of the RPC servers associated with system services in Microsoft Exchange Server, SQL Server, Windows NT 4.0 and Windows 2000 do not adequately validate inputs, and in some cases will accept invalid inputs that prevent normal processing. The specific input values at issue here vary from RPC server to RPC server. An attacker who sent such inputs to an affected RPC server could disrupt its service. The precise type of disruption would depend on the specific service, but could range in effect from minor (e.g., the service temporarily hanging) to major (e.g., the service failing in a way that would require the entire system to be restarted)."
6. ^ Mark Russinovich (October 1997). *Inside NT's Object Manager*. Introduction.
7. ^ Mark Russinovich (October 1997). *Inside NT's Object Manager*. "Object Types".
8. ^ Mark Russinovich (October 1997). *Inside NT's Object Manager*. "Objects".
9. ^ *Inside Microsoft Windows 2000* (Third Edition). *Microsoft Press.*

References

- Finnel, Lynn (2000). *MCSE Exam 70-215, Microsoft Windows 2000 Server*. Chapter 1, Introduction to Microsoft Windows 2000, pg 7-18. Microsoft Press. ISBN 1-57231-903-8.
- *Microsoft*. Running Nonnative Applications in Windows 2000 Professional. Windows 2000 Resource Kit. Retrieved May

4, 2005.
- *Microsoft*. "Active Directory Data Storage". Retrieved May 9, 2005.
- Russinovich, Mark (October 1997). "Inside NT's Object Manager". *Windows IT Pro.*
- Siyan, Kanajit S. (2000). "Windows 2000 Professional Reference". *New Riders*. ISBN 0735709521.
- Salomon, David; & Russinovich, Mark E. (2000). *Inside Microsoft Windows 2000* (Third Edition). *Microsoft Press*. ISBN 0735610215.

See also

- Comparison of operating systems

External links

- Official Page
- Windows 2000 Server comparison chart
- GUIdebook: Windows 2000 Gallery - A website dedicated to preserving and showcasing Graphical User Interfaces

History of Microsoft Windows

Windows: 1.0 | 2.0 | 3.x | NT | 95 | 98 | Me | **2000** | XP | Server 2003 | CE | Mobile | Longhorn | Blackcomb

Mentioned In

windows 2000 is mentioned in the following topics:

QOS Packet Scheduler (technology)	Windows 95/98/NT (technology)
My Network Places (technology)	Terminal Services (technology)
SBS (technology)	Cairo (technology)
Microsoft Narrator	Martin M. Weiss
domain controller	EFS (IRIX)
	More>

Shopping

Microsoft Windows 2000

Professional (B23-00079) for PC

CDW

Store Rating

★★★½

$276.97 BUY NOW

COMPARE PRICES

Microsoft Windows 2000

Professional Full Version (b23-00079-oem)

VioSOFTWARE

Store Rating

★★★★½

$128.00 BUY NOW

COMPARE PRICES

Microsoft Windows 2000

Professional Full Version (B23-00079-RB) for PC

VioSOFTWARE

Store Rating

★★★★½

$197.44 BUY NOW

COMPARE PRICES

Narrow Your Search

Brand

- Microsoft
- IBM

Shop

Enter a keyword (browse)

Choose a category

TRIVIA MACHINE

What are the minimum system requirements to run Windows 2000?

- 133 Mhz Processor, 64 Mb Ram, 650 Mb Free Space
- 233 Mhz Processor, 64 Mb Ram, 750 Mb Free Space
- 233 Mhz Processor, 128 Mb Ram, 850 Mb Free Space
- 133 Mhz Processor, 32 Mb Ram, 550 Mb Free Space

Find out

Play Trivia Machine